

1 CLAIMS

2 1. A method of factoring operating system functions comprising:
3 defining criteria that governs how functions of an operating system are to
4 be factored into one or more groups;
5 factoring the functions into one or more groups based upon the criteria; and
6 associating groups of functions with programming objects that have data
7 and methods, wherein the methods correspond to the operating system functions.
8

9 2. The method of claim 1, wherein the programming objects have
10 interfaces through which the methods can be accessed.
11

12 3. The method of claim 1, wherein the programming objects comprise
13 COM objects.
14

15 4. The method of claim 1, wherein said factoring comprises creating a
16 hierarchy of object interfaces in which certain interfaces can inherit from other
17 interfaces.
18

19 5. The method of claim 1, wherein said factoring comprises creating a
20 hierarchy of object interfaces in which certain interfaces can aggregate with other
21 interfaces.
22

23 6. The method of claim 1 further comprising instantiating a plurality of
24 programming objects across a process boundary.
25

1 7. The method of claim 1, further comprising instantiating a plurality of
2 programming objects across a machine boundary.

3
4 8. The method of claim 1, wherein the criteria is based, at least in part,
5 on the manner in which particular functions behave.

6
7 9. The method of claim 8, wherein the manner includes a consideration
8 of the types of operating system resources that are associated with the operation of
9 a function.

10
11 10. The method of claim 8, wherein the manner includes a consideration
12 of whether a particular function creates an operating system resource.

13
14 11. The method of claim 8, wherein the manner includes a consideration
15 of whether a particular function operates upon an operating system resource.

16
17 12. The method of claim 1, wherein the criteria is based, at least in part,
18 on the manner in which particular functions behave, wherein the manner includes:

19 a consideration of the types of operating system resources that are
20 associated with the operation of a function; and

21 a consideration of whether a particular function creates an operating system
22 resource.

1 **13.** The method of claim 1, wherein the criteria is based, at least in part,
2 on the manner in which particular functions behave, wherein the manner includes:
3 a consideration of the types of operating system resources that are
4 associated with the operation of a function call; and
5 a consideration of whether a particular function operates upon a given
6 operating system resource.

7
8 **14.** A method of factoring operating system functions comprising:
9 factoring a plurality of operating system functions that are used in
10 connection with operating system resources into first groups based upon first
11 criteria;
12 factoring the first groups into individual sub-groups based upon second
13 criteria; and
14 assigning each sub-group to its own programming object interface, wherein
15 a programming object interface represents a particular object's implementation of
16 its collective methods.

17
18 **15.** The method of claim 14, wherein the first criteria is based upon the
19 type of resource that is associated with an operation of a function.
20

21 **16.** The method of claim 14, wherein the second criteria is based upon
22 the nature of an operation of a function on a particular resource.
23
24
25

1 **17.** The method of claim 16, wherein said nature concerns whether a
2 function creates a resource.

3
4 **18.** The method of claim 16, wherein said nature concerns whether a
5 function does not create a resource.

6
7 **19.** The method of claim 14, wherein the first criteria is based upon the
8 type of resource that is associated with an operation of a function, and the second
9 criteria is based upon the nature of an operation of a function on a particular
10 resource.

11
12 **20.** The method of claim 14, wherein at least one interface inherits from
13 another interface.

14
15 **21.** The method of claim 14, wherein at least one interface aggregates
16 with another interface.

17
18 **22.** The method of claim 14 further comprising instantiating a plurality
19 of programming objects across a process boundary.

20
21 **23.** The method of claim 14 further comprising instantiating a plurality
22 of programming objects across a process boundary and a machine boundary.
23
24
25

1 **24.** A method of factoring operating system functions comprising:
2 factoring a plurality of operating system functions into interface groups
3 based upon the resources with which a function is associated;
4 factoring the interface groups into interface sub-groups based upon each
5 function's use a handle that represents a resource; and
6 organizing the interface sub-groups so that at least one of the interface sub-
7 groups inherits from at least one other of the interface sub-groups.

8
9 **25.** The method of claim 24, wherein said organizing comprises
10 aggregating at least one of the interface sub-groups.

11
12 **26.** The method of claim 24, wherein the interface sub-groups are
13 associated with COM objects.

14
15 **27.** The method of claim 24, wherein the factoring of the interface
16 groups into interface sub-groups comprises considering whether a function creates
17 a handle.

1 **28.** The method of claim 24, wherein said organizing comprises
2 aggregating at least one of the interface sub-groups, and wherein the factoring of
3 the interface groups into interface sub-groups comprises considering whether a
4 function call creates a handle.

5
6 **29.** An operating system application program interface embodied on a
7 computer-readable medium comprising a plurality of object interfaces, wherein
8 each object interface includes one or more methods that are associated with and
9 can call functions of an operating system that does not comprise the object
10 interfaces.

11
12 **30.** The operating system application program interface of claim 29,
13 wherein the object interfaces are arranged in groups in accordance with the types
14 of objects with which their operation is associated.

15
16 **31.** The operating system application program interface of claim 29,
17 wherein the methods within some of the interfaces are arranged in accordance with
18 whether they create an object.

19
20 **32.** The operating system application program interface of claim 29,
21 wherein the methods within some of the interfaces are arranged in accordance with
22 whether they do not create an object.

23
24
25

1 **33.** The operating system application program interface of claim 29,
2 wherein the methods within some of the interfaces are arranged in accordance with
3 whether they operate upon an object.
4

5 **34.** The operating system application program interface of claim 29,
6 wherein at least some of the object interfaces are arranged so that they inherit from
7 other of the object interfaces.
8

9 **35.** The operating system application program interface of claim 29,
10 wherein at least some of the object interfaces are arranged so that they aggregate
11 with other of the object interfaces.
12

13 **36.** An operating system comprising:
14 a plurality of programming objects having interfaces, wherein the
15 programming objects represent operating system resources, and wherein the
16 interfaces define methods that are organized in accordance with whether they
17 create an operating system resource or not;

18 wherein the programming objects are configured to be called either directly
19 or indirectly by an application; and

20 wherein the methods are configured to call operating system functions
21 responsive to being called directly or indirectly by an application.
22

23 **37.** The operating system of claim 36, wherein some of the objects are
24 disposed across at least one process boundary.
25

1 38. The operating system claim 36, wherein some of the objects are
2 disposed across at least one machine boundary.

3
4 39. The operating system application program interface of claim 36,
5 wherein at least some of the objects are disposed across at least one process
6 boundary and at least one machine boundary.

7
8 40. The operating system application program interface of claim 36,
9 wherein the objects comprise COM objects.

10
11 41. A method of converting an operating system from a non-object-
12 oriented format to an object oriented format, wherein the operating system
13 includes a plurality of operating system functions that are callable to create or use
14 operating system resources, the method comprising:

15 defining a plurality of programming object interfaces that define methods
16 that correspond to the operating system functions, wherein programming objects
17 that support the interfaces are callable either directly or indirectly by an
18 application;

19 calling a programming object interface; and
20 responsive to said calling, calling an operating system function with a
21 method of the programming object that supports said programming object
22 interface.